Detecting Stuffing of a User's Credentials at Her Own Accounts

Ke Coby Wang       Michael K. Reiter

University of North Carolina at Chapel Hill
Credential Stuffing

Database breaches, phishing, malware, social engineering, etc.

Valid user ID
password pairs
Harm of Credential Stuffing

Leaked passwords

Password reuse

Credential stuffing

Account takeovers

Web service providers:
- Stolen credential ransom
- Costs for preventing & detecting account takeovers
- Customer churn

Web service users:
- Financial loss
- Privacy violations

...
Existing Approaches

Pre-attack

- Leaked passwords
- Password reuse

Attack

- Credential stuffing

Post-attack

- Account takeovers

Detecting & cross-checking leaked passwords
Existing Approaches

Detecting & cross-checking leaked passwords

Pre-attack

Leaked passwords

Password reuse

Detecting & discouraging password reuse

Attack

Credential stuffing

Post-attack

Account takeovers
Existing Approaches

Detecting & cross-checking leaked passwords

Pre-attack

Leaked passwords

Password reuse

Attack

Credential stuffing

Post-attack

Account takeovers

Compromised account detection, account activity monitoring, etc.

Detecting & discouraging password reuse
Existing Approaches

**Pre-attack**
- Leaked passwords
- Password reuse

**Attack**
- Adding more authentication factors: 2FA, MFA, etc.
- Credential stuffing

**Post-attack**
- Account takeovers
- Honey accounts, account activity monitoring, etc.

Detecting & cross-checking leaked passwords
Detecting & discouraging password reuse
Existing Approaches

- **Pre-attack**
  - Leaked passwords
  - Password reuse

- **Attack**
  - Credential stuffing
  - How to directly detect?
    - Adding more authentication factors: 2FA, MFA, etc.

- **Post-attack**
  - Account takeovers
    - Honey accounts, account activity monitoring, etc.

Detecting & cross-checking leaked passwords
Detecting & discouraging password reuse
Our Work

**Pre-attack**
- Leaked passwords
- Password reuse

**Attack**
- Adding more authentication factors: 2FA, MFA, etc.
- Credential stuffing

**Post-attack**
- Account takeovers
- Honey accounts, account activity monitoring, etc.

**Our work:**
- Detecting active credential stuffing across multiple websites
- Detecting & cross-checking leaked passwords
- Detecting & discouraging password reuse
Anomaly Detection Systems (ADS)

User login histories (e.g., Login IP addresses, useragent strings, device fingerprints)

This attempt is **Abnormal**.

?  →  Login attempt  →  ADS at Website  →  Decision 1, Decision 2

This attempt is **normal**.
Anomaly Detection Systems (ADS)

User login histories (e.g., Login IP addresses, userAgent strings, device fingerprints)

Naïve ADS:
- Strange IPs = “abnormal”
- Strange devices = “abnormal”
Anomaly Detection Systems (ADS)

User login histories (e.g., Login IP addresses, useragent strings, device fingerprints)

Login attempt

? → ADS at Website

Decision 1

This attempt is Abnormal.

Decision 2

This attempt is normal.

More sophisticated ADS*:

- Multiple login features
- Attackers’ different capability levels

*Freeman et al. [NDSS 2016]
Anomaly Detection Systems (ADS)

User login histories (e.g., Login IP addresses, useragent strings, device fingerprints)

```
?  Login attempt

ADS at Website

Decision 1
This attempt is Abnormal.

Decision 2
This attempt is normal.
```

“Researching attacker”*:
- Hold users’ correct passwords
- Try to access users’ accounts from same countries of legitimate users

“Phishing attacker”*:
- Hold users’ correct passwords
- Try to access users’ accounts from same countries with same browser user-agent strings of legitimate users

*Freeman et al. [NDSS 2016]
Anomaly Detection Systems (ADS)

User login histories (e.g., Login IP addresses, useragent strings, device fingerprints)

This attempt is **Abnormal**.

This attempt is **normal**.

**ADS:**
- leverages users’ login patterns (IPs, browser agentstrings, etc.)
- helps a website to distinguish malicious login attempts
- **NOT** an authentication factor that directly decides whether a login attempt is successful or not.
Evidence Trail from Credential Stuffing

c = “alice@yyy.com : alicepwd”,
a leaked username-password pair possessed by the credential stuffer

Websites where Alice has accounts

- alice@yyy.com : alicepwd0
  - ADS
  - 2FA
- alice@yyy.com : alicepwd
  - ADS
  - 2FA
- alice@yyy.com : alicepwd
  - ADS

Credential Stuffer
Evidence Trail from Credential Stuffing

c = “alice@yyy.com : alicepwd”, a leaked username-password pair possessed by the credential stuffer

Websites where Alice has accounts

alice@yyy.com : alicepwd0
ADS: abnormal

Login attempt with c

Credential Stuffer

alice@yyy.com : Alicepwd

≠ alicepwd

2FA

ADS

2FA

ADS
Evidence Trail from Credential Stuffing

c = “alice@yyy.com : alicepwd”, a leaked username-password pair possessed by the credential stuffer

Websites where Alice has accounts

Login attempt with c

Credential Stuffer

= alicepwd

≠ alicepwd

ADS : abnormal

2FA : failed

ADS : abnormal

ADS
Evidence Trail from Credential Stuffing

c = “alice@yyy.com : alicepwd”, a leaked username-password pair possessed by the credential stuffer

Websites where Alice has accounts

Login attempt with c

Accessibility Services: abnormal

Login attempt with c

2FA: failed

Login attempt with c

Accounts: abnormal

Credential Stuffer
The “trail” left by credential stuffing attacks are those passwords submitted in abnormal login attempts that fail:

- **Without 2FA**
  - ADS reports “abnormal”; the submitted password is incorrect

- **With 2FA:**
  - ADS reports “abnormal”; the submitted password is incorrect
  - ADS reports “abnormal”; the submitted password is correct but 2FA fails
Our Framework

c = “alice@yyy.com : \textit{alicepwd}”, a leaked username-password pair possessed by the credential stuffer

Websites where Alice has accounts

- alice@yyy.com : \textit{alicepwd0}
- alice@yyy.com : \textit{alicepwd}
- alice@yyy.com : \textit{alicepwd}

Credential Stuffer
Our Framework

c = “alice@yyy.com : alicepwd”, a leaked username-password pair possessed by the credential stuffer

Websites where Alice has accounts

- alice@yyy.com : alicepwd0
  - ADS: abnormal
  - SUSPICIOUS: {}

- alice@yyy.com : alicepwd
  - ADS
  - 2FA

- alice@yyy.com : alicepwd
  - ADS

Credential Stuffer

Login attempt with c
Our Framework

c = “alice@yyy.com : alicepwd”, a leaked username-password pair possessed by the credential stuffer

Login attempt with c

Credential Stuffer

Websites where Alice has accounts

alice@yyy.com : alicepwd0
ADS: abnormal
SUSPICIOUS: { alicepwd }

alice@yyy.com : alicepwd
ADS
2FA

alice@yyy.com : alicepwd
ADS
2FA
Our Framework

c = “alice@yyy.com : alicepwd”, a leaked username-password pair possessed by the credential stuffer

Websites where Alice has accounts

alice@yyy.com : alicepwd
ADS: abnormal
SUSPICIOUS: { alicepwd }

Login attempt with c

2FA

alice@yyy.com : alicepwd = alicepwd
ADS: abnormal
2FA: failed
SUSPICIOUS: { alicepwd }

collect
Our Framework

$c = "alice@yyy.com : alicepwd", a leaked username-password pair possessed by the credential stuffer$

Websites where Alice has accounts

- alice@yyy.com : alicepwd0
  - ADS: abnormal
  - SUSPICIOUS: { alicepwd }

- alice@yyy.com : alicepwd
  - ADS: abnormal
  - 2FA: failed
  - SUSPICIOUS: { alicepwd }

- alice@yyy.com : alicepwd
  - ADS

COLLECTING phase

Credential Stuffer

Login attempt with c

≠ alicepwd

collect

= alicepwd

collect

THE UNIVERSITY OF NORTH CAROLINA AT CHAPEL HILL
c = “alice@yyy.com : alicepwd”, a leaked username-password pair possessed by the credential stuffer

Websites where Alice has accounts

- alice@yyy.com : alicepwd0
  ADS: abnormal
  SUSPICIOUS: {alicepwd}

- alice@yyy.com : alicepwd
  ADS: abnormal
  2FA: failed
  SUSPICIOUS: {alicepwd}

- alice@yyy.com : alicepwd
  ADS: abnormal
  = alicepwd

Credential Stuffer

Login attempt with c
Our Framework

c = “alice@yyy.com : alicepwd”, a leaked username-password pair possessed by the credential stuffer

Websites where Alice has accounts

- alice@yyy.com : alicepwd0
  - ADS: abnormal
  - SUSPICIOUS: { alicepwd }

- alice@yyy.com : alicepwd
  - 2FA: failed
  - SUSPICIOUS: { alicepwd }

Have you collected “alicepwd” for “alice@yyy.com”?

Login attempt with c:

- alice@yyy.com : alicepwd
  - ADS: abnormal
  - = alicepwd
Our Framework

c = “alice@yyy.com : alicepwd”, a leaked username-password pair possessed by the credential stuffer.

Websites where Alice has accounts:

- alice@yyy.com : alicepwd0
  - ADS: abnormal
  - SUSPICIOUS: { alicepwd }

- alice@yyy.com : alicepwd
  - ADS: abnormal
  - 2FA: failed
  - SUSPICIOUS: { alicepwd }

- alice@yyy.com : alicepwd
  - ADS: abnormal
  - 2FA

A positive detection happens when the number of received positive responses is $\geq$ a pre-set threshold ("attack width").
Our Framework

c = “alice@yyy.com : alicepwd”, a leaked username-password pair possessed by the credential stuffer

Websites where Alice has accounts

- alice@yyy.com : alicepwd
  - ADS: abnormal
  - 2FA: failed
  - SUSPICIOUS: { alicepwd }

- alice@yyy.com : alicepwd0
  - ADS: abnormal

Have you collected “alicepwd” for “alice@yyy.com”?

Login attempt with c

COUNTING phase
Two important questions:

• **False detection rate (FDR)**
  - *What if a (forgetful) user "guesses" her own passwords at her accounts?*

• **True detection rate (TDR)**
  - *What if a credential stuffer tries to circumvent detection by trying a smart attack strategy?*
Conservatively Estimating FDR & TDR

- **A forgetful user as a MDP**:  
  - Maximizing the probability of triggering a false detection (false detection rate)
Conservatively Estimating FDR & TDR

- A forgetful user as a MDP*:
  - Maximizing the probability of triggering a false detection (false detection rate)

- A credential stuffer as a MDP*:
  - Minimizing the probability of getting detected while maximizing the number of account takeovers (true detection rate)

* MDP: Markov decision process
Conservatively Estimating FDR & TDR

*Phishing attackers*: valid passwords from same countries with same browser user-agent strings of legitimate users

* Freeman et al. (NDSS 2016)
Conservatively Estimating FDR & TDR

**Phishing attackers**: valid passwords from same countries with same browser user-agent strings of legitimate users

- **Default (baseline) setting**: some level of password reuse in a set of 4 distinct passwords across 10 accounts (one per site) with no 2FA deployed among them

* Freeman et al. (NDSS 2016)
Conservatively Estimating FDR & TDR

**Phishing attackers**: valid passwords from same countries with same browser user-agent strings of legitimate users

- **Default (baseline) setting**: some level of password reuse in a set of 4 distinct passwords across 10 accounts (one per site) with no 2FA deployed among them
- **Blue curves**: each for a different ADS threshold in the collecting phase

*Freeman et al. (NDSS 2016)*
Conservatively Estimating FDR & TDR

Phishing attackers*: valid passwords from same countries with same browser user-agent strings of legitimate users

- **Default (baseline) setting**: some level of password reuse in a set of 4 distinct passwords across 10 accounts (one per site) with no 2FA deployed among them
- **Blue curves**: each for a different ADS threshold in the collecting phase
- **Black, dashed curves**: corresponding ADS’s accuracy in detecting abnormal logins

* Freeman et al. (NDSS 2016)
Phishing attackers*: valid passwords from same countries with same browser user-agent strings of legitimate users

- **Default (baseline) setting**: some level of password reuse in a set of 4 distinct passwords across 10 accounts (one per site) with no 2FA deployed among them
- **Blue curves**: each for a different ADS threshold in the collecting phase
- **Black, dashed curves**: corresponding ADS’s accuracy in detecting suspicious logins
- **Black, dotted lines**: random guessing

Baseline

* Freeman et al. (NDSS 2016)
Conservatively Estimating FDR & TDR

**Phishing attackers**: valid passwords from same countries with same browser user-agent strings of legitimate users

- **Default (baseline) setting**: some level of password reuse in a set of 4 distinct passwords across 10 accounts (one per site) with no 2FA deployed among them
- **Blue curves**: each for a different ADS threshold in the collecting phase
- **Black, dashed curves**: corresponding ADS’s accuracy in detecting suspicious logins
- **Black, dotted lines**: random guessing

```
 Baseline
 Less pwd reuse  # of pwds +1  # of accnts + 10  # of 2FA + 5  Higher ADS detection rates in the counting phase
```

* Freeman et al. (NDSS 2016)
Conservatively Estimating FDR & TDR

**Phishing attackers**: valid passwords from same countries with same browser user-agent strings of legitimate users

- **Default (baseline) setting**: some level of password reuse in a set of 4 distinct passwords across 10 accounts (one per site) with no 2FA deployed among them
- **Blue curves**: each for a different ADS threshold in the collecting phase
- **Black, dashed curves**: corresponding ADS’s accuracy in detecting suspicious logins
- **Black, dotted lines**: random guessing

Baseline | Less pwd reuse | # of pwds +1 | # of accnts +10 | # of 2FA +5 | Higher ADS detection rates in the counting phase

* Freeman et al. (NDSS 2016)
Conservatively Estimating FDR & TDR

**Phishing attackers**: valid passwords from same countries with same browser user-agent strings of legitimate users

- **Default (baseline) setting**: some level of password reuse in a set of 4 distinct passwords across 10 accounts (one per site) with no 2FA deployed among them
- **Blue curves**: each for a different ADS threshold in the collecting phase
- **Black, dashed curves**: corresponding ADS’s accuracy in detecting suspicious logins
- **Black, dotted lines**: random guessing

Baseline | Less pwd reuse | # of pwds +1 | # of accnts + 10 | # of 2FA + 5 | Higher ADS detection rates in the counting phase

* Freeman et al. (NDSS 2016)
**Conservatively Estimating FDR & TDR**

**Phishing attackers**: valid passwords from same countries with same browser user-agent strings of legitimate users

- **Default (baseline) setting**: some level of password reuse in a set of 4 distinct passwords across 10 accounts (one per site) with no 2FA deployed among them
- **Blue curves**: each for a different ADS threshold in the collecting phase
- **Black, dashed curves**: corresponding ADS’s accuracy in detecting suspicious logins
- **Black, dotted lines**: random guessing

<table>
<thead>
<tr>
<th>Baseline</th>
<th>Less pwd reuse</th>
<th># of pwds +1</th>
<th># of accnts +10</th>
<th># of 2FA +5</th>
<th>Higher ADS detection rates in the counting phase</th>
</tr>
</thead>
</table>

* Freeman et al. (NDSS 2016)
Researching attackers*: valid passwords from same countries of legitimate users.

Baseline | Less pwd reuse | # of pwds +1 | # of accnts + 10 | # of 2FA + 5 | Higher ADS detection rates in the counting phase

* Freeman et al. (NDSS 2016)
Other features of our framework:
Other features of our framework:

- *Account security*
**Account Security**

**COUNTING phase**

**Have you collected**
“**alicepwd**” for “alice@yyy.com”?

**Attacker**
Login attempt with c

**Websites where Alice has accounts**

- alice@yyy.com : **alicepwd0**
  - ADS: abnormal
  - SUSPICIOUS: { **alicepwd** }

- alice@yyy.com : **alicepwd**
  - ADS: abnormal
  - 2FA: failed
  - SUSPICIOUS: { **alicepwd** }

- alice@yyy.com : **alicepwd**
  - ADS: abnormal

**c = “alice@yyy.com : alicepwd”**
Account Security

**COUNTING phase**

Websites where Alice has accounts

- alice@yyy.com : alicepwd0
  - ADS: abnormal
  - SUSPICIOUS: { alicepwd }
- alice@yyy.com : alicepwd
  - ADS: abnormal
  - 2FA: failed
  - SUSPICIOUS: { alicepwd }
- alice@yyy.com : alicepwd
  - ADS: abnormal

**Private membership test (PMT) query**

Have you collected “alicpwd” for “alice@yyy.com”?  

**Attacker**

Login attempt with c

\[c = \text{“alice@yyy.com : alicepwd”}\]
Other features of our framework:

- **Account security**
  - *A new one-round two-party private membership test (PMT) protocol*
Other features of our framework:

- **Account security**
  - A new one-round two-party private membership test (PMT) protocol

- **Directory**
Other features of our framework:

- **Account security**
  - A new one-round two-party private membership test (PMT) protocol

- **Directory**
  - A “look-up table” that maintains where a user has accounts
Directory

**COUNTING phase**

Websites where Alice has accounts

- alice@yyy.com : alicepwd0
  - ADS: abnormal
  - SUSPICIOUS: { alicepwd }

- alice@yyy.com : alicepwd
  - ADS: abnormal
  - 2FA: failed
  - SUSPICIOUS: { alicepwd }

- alice@yyy.com : alicepwd
  - ADS: abnormal

How to find?

Attacker

Login attempt with c

\[ c = \text{“alice@yyy.com : alicepwd”} \]
Directory

\[ ... \]

\[ alice@yyy.com: \]
Site #1, #2, ...

\[ ... \]

\[ bob@zzz.com: \]
Site #1, #3, ...

\[ ... \]

\[ COUNTING phase \]

Websites where Alice has accounts

\[ alice@yyy.com : alicepwd0 \]
ADS: abnormal
SUSPICIOUS: \{ alicepwd \}

\[ 2FA \]
alice@yyy.com : alicepwd
ADS: abnormal
2FA: failed
SUSPICIOUS: \{ alicepwd \}

\[ 2FA \]
alice@yyy.com : alicepwd
ADS: abnormal

\[ c = \text{"alice@yyy.com : alicepwd"} \]
Directory

... 
alice@yyy.com: Site #1, #2, ... 
... bob@zzz.com: Site #1, #3, ... 
...

COUNTING phase

Websites where Alice has accounts

- alice@yyy.com: alicepwd0
  - ADS: abnormal
  - SUSPICIOUS: { alicepwd }

- alice@yyy.com: alicepwd
  - ADS: abnormal
  - 2FA: failed
  - SUSPICIOUS: { alicepwd }

- alice@yyy.com: alicepwd
  - ADS: abnormal

2FA

Login attempt with c

c = “alice@yyy.com : alicepwd”
Directory

\[c = \text{"alice@yyy.com : alicepwd"}\]
Directory

**COUNTING phase**

Websites where Alice has accounts

- alice@yyy.com : alicepwd0
  - ADS: abnormal
  - SUSPICIOUS: { alicepwd }

- alice@yyy.com : alicepwd
  - ADS: abnormal
  - 2FA: failed
  - SUSPICIOUS: { alicepwd }

- alice@yyy.com : alicepwd
  - ADS: abnormal

**Attacker**

Login attempt with c

\[ c = \text{“alice@yyy.com : alicepwd”} \]
Other features of our framework:

- **Account security**
  - A new one-round two-party private membership test (PMT) protocol

- **Directory**
  - A “look-up table” that maintains where a user has accounts

- **Login privacy**
Other features of our framework:

- **Account security**
  - A new one-round two-party private membership test (PMT) protocol

- **Directory**
  - A “look-up table” that maintains where a user has accounts

- **Login privacy**
  - *Trusted directory for login privacy*
Login Privacy

**COUNTING phase**

**Responder**
- Websites where Alice has accounts
  - alice@yyy.com : alicepwd0
  - ADS: abnormal
  - SUSPICIOUS: { alicepwd }

**User**
- Login attempt
- Requester
  - alice@yyy.com : alicepwd
  - ADS: abnormal
  - 2FA: failed
  - SUSPICIOUS: { alicepwd }

**Trusted Directory**
- alice@yyy.com : alicepwd
- ADS: abnormal
- 2FA: failed
- SUSPICIOUS: { alicepwd }
Other features of our framework:

- **Account security**
  - A new one-round two-party private membership test (PMT) protocol

- **Directory**
  - A “look-up table” that maintains where a user has accounts

- **Login privacy**
  - Trusted directory for login privacy
  - *Untrusted directory for login privacy*
Login Privacy

COUNTING phase

Websites where Alice has accounts

Untrusted Directory

Tor

User

Login attempt

SUSPICIOUS: { alicepwd }

alice@yyy.com : alicepwd0
ADS: abnormal

alice@yyy.com : alicepwd
ADS: abnormal
2FA: failed

SUSPICIOUS: { alicepwd }

alice@yyy.com : alicepwd
ADS: abnormal
Scalability

Max. qualifying responses per sec.

\[ 2^7 \quad 2^8 \quad 2^9 \quad 2^{10} \quad \text{Susp. set size at responders} \]

Number of responders

<table>
<thead>
<tr>
<th>Number of responders</th>
<th>25</th>
<th>50</th>
<th>75</th>
<th>100</th>
<th>125</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 2^7 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 2^8 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 2^9 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 2^{10} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Trusted** directory for login privacy
  - (Qualifying response: \( \leq 5s \))
- **Untrusted** directory for login privacy
  - (Qualifying response: \( \leq 8s \))
Scalability

Max. qualifying responses per sec.

Number of responders

- **Trusted** directory for login privacy
  - (Qualifying response: <= 5s)

- **Untrusted** directory for login privacy
  - (Qualifying response: <= 8s)

**Susp. set size at responders**

Graphs showing scalability with different responder counts and response times.
Scalability

Max. qualifying responses per sec.

Number of responders

Trust\textit{ed} directory for login privacy

Untrusted directory for login privacy

(Qualifying response: <= 5s) (Qualifying response: <= 8s)
Scalability

Max. qualifying responses per sec.

Response time measured at the requester
Scalability

Max. qualifying responses per sec.

**Trusted** directory for login privacy
(Qualifying response: <= 5s)

**Untrusted** directory for login privacy
(Qualifying response: <= 8s)
Scalability

Max. qualifying responses per sec.

Trusted directory for login privacy
(Qualifying response: <= 5s)

Untrusted directory for login privacy
(Qualifying response: <= 8s)
Scalability

Max. qualifying responses per sec.

\[ \text{Susp. set size at responders} \]

\[ 2^7 \quad 2^8 \quad 2^9 \quad 2^{10} \]

Number of responders

- **Trusted** directory for login privacy (Qualifying response: \( \leq 5s \))
- **Untrusted** directory for login privacy (Qualifying response: \( \leq 8s \))
## Scalability

<table>
<thead>
<tr>
<th></th>
<th>Credential-stuffing login attempts per day</th>
<th>Proportion that succeed</th>
<th>Proportion of all login attempts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airline</td>
<td>1.4 Million</td>
<td>1.00%</td>
<td>60%</td>
</tr>
<tr>
<td>Hotel</td>
<td>4.3 Million</td>
<td>1.00%</td>
<td>44%</td>
</tr>
<tr>
<td>Retail</td>
<td>131.5 Million</td>
<td>0.50%</td>
<td>91%</td>
</tr>
<tr>
<td>Consumer banking</td>
<td>232.2 Million</td>
<td>0.05%</td>
<td>58%</td>
</tr>
</tbody>
</table>

Table: Credential stuffing estimates for four major U.S. industries*

**Total number of PMT queries per second:**

- If ADS false & true detection rates are 0.30 & 0.95 (against phishing attackers): **660**
- If ADS false & true detection rates are 0.10 & 0.99 (against researching attackers): **227**

* Shape Security, “2018 Credential spill report”
Scalability

Max. qualifying responses per sec.

Number of responders

Trusted directory for login privacy

Untrusted directory for login privacy

Susp. set size at responders

2^7

2^8

2^9

2^10

65 responders

32 responders

660 (phishing attackers)

227 (researching attackers)
Summary

- A framework to detect credential stuffing
Summary

- A framework to detect credential stuffing
  - *Leverages ADS and evidence trail left by credential stuffing*
Summary

- A framework to detect credential stuffing
  - Leverages ADS and evidence trail left by credential stuffing
  - *Account security achieved by a novel PMT protocol*
Summary

- A framework to detect credential stuffing
  - Leverages ADS and evidence trail left by credential stuffing
  - Account security achieved by a novel PMT protocol
  - *Login privacy enforced by the directory or by Tor*
Summary

- A framework to detect credential stuffing
  - Leverages ADS and evidence trail left by credential stuffing
  - Account security achieved by a novel PMT protocol
  - Login privacy enforced by the directory or by Tor

- *First to detect active credential stuffing attacks across multiple websites*
Summary

- A framework to detect credential stuffing
  - Leverages ADS and evidence trail left by credential stuffing
  - Account security achieved by a novel PMT protocol
  - Login privacy enforced by the directory or by Tor
- First to detect active credential stuffing attacks across multiple websites
- *Even a minimal-infrastructure deployment of our framework should support the combined login load experienced by four major sectors of the U.S economy*
Thank you!

Coby Wang
Email: kwang@cs.unc.edu